KATHOLIEKE
UNIVERSITEIT
LEUVEN

DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

RESEARCH REPORT 9909

PROVABLE BOUNDS FOR THE MEAN QUEUE LENGTHS IN A HETEROGENEOUS PRIORITY QUEUE

by
H. LEEMANS

D/1999/2376/09

Provable Bounds for the Mean Queue Lengths in a Heterogeneous Priority Queue

H. Leemans*

Abstract

We analyze a two-class two-server system with nonpreemptive heterogeneous priority structures. We use matrix-geometric techniques to determine the stationary queue length distributions. Numerical solution of the matrix-geometric model requires that the number of phases be truncated and it is shown how this affects the accuracy of the results. We then establish and prove upper and lower bounds for the mean queue lengths under the assumption that the classes have equal mean service times.

1 Introduction

Queues with nonpreemptive heterogeneous priorities generally arise in batch job processing within MVS mainframe environments. Batch jobs are divided into job classes, based upon their resource requirements (e.g. cpu seconds, number of tape units required, memory requirements, ...) and executed in separate batch address spaces, called *initiators*. Several initiators may be active at a time; this makes it possible to process multiple batch jobs in parallel.

The number of initiators has to be defined by the system performance manager. This definition includes a list of the job classes they shall execute. A simple initiator definition example, with only five active initiators, is shown in Figure 1. The first initiator (I1) is a single class initiator, dedicated to class X jobs. Initiators I2 and I3 execute both class A and class B jobs. The order in which the classes are listed imposes a priority structure on these classes. Therefore, class A has priority over class B on both initiators; such priority structures are called *homogeneous*. Initiators I4 and I5 process

*K.U.Leuven, Department of Applied Economic Sciences, Naamsestraat 69, B-3000 Leuven, Belgium. Email: herlinde.leemans@econ.kuleuven.ac.be

jobs of classes C and D each. Here, I4 executes class C jobs before class D jobs, whereas I5 gives priority to class D; these initiators have *heterogeneous* priority structures. The priorities are *nonpreemptive*; e.g. a newly arriving class D job cannot preempt a class C job being executed by I5.

```
INITDEF   PARTNUM=5
INIT001   CLASS=X, START, NAME=I1
INIT002   CLASS=AB, START, NAME=I2
INIT003   CLASS=AB, START, NAME=I3
INIT004   CLASS=CD, START, NAME=I4
INIT005   CLASS=DC, START, NAME=I5
```

Figure 1: Initiator initialization code: an example.

The queues represented by initiators I1 through I3 are well-known. In this paper, we focus on the analysis of queues as I4 and I5, for which we first give a formal definition below.

Consider a system consisting of two servers (S_1 and S_2) and two job classes (A and B) as illustrated in Figure 2. Class A has nonpreemptive priority over class B on S_1; class B has nonpreemptive priority over class A on S_2. Both classes have Poisson arrivals with parameters λ_a and λ_b respectively. Service times are exponentially distributed with average $1/\mu_a$ and $1/\mu_b$ and μ_a may differ from μ_b. The servers select jobs for service depending on the state of the queues: S_1 will only select a class B job if the class A queue is empty, otherwise it selects class A jobs. S_2 selects class A jobs only if the queue of class B is empty, otherwise it selects class B jobs. If both servers are idle, an arriving job is processed by the server which offers the highest priority. Service discipline within each class is FCFS. We shall only analyze stable systems, for which $\rho_a + \rho_b < 2$ (for a proof of this condition, see Leemans [3]).

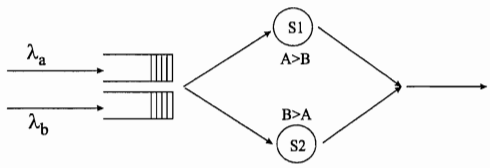

Figure 2: Two-class two-server priority queueing model with heterogeneous priority structures.

In the following section, we shall describe a matrix-geometric model suited to compute the joint and stationary queue length distributions and their

moments. Numerical solution requires that the state space be truncated and we show, using numerical experiments in Section 3, that this leads to lower bounds for the average queue lengths of both classes. In Section 4, we prove these bounds using the precedence relation technique (van Houtum [5]) for the case where the classes have equal mean service times. Next, using the same technique, we derive upper bounds as well. However, the technique fails for the case where the classes have unequal mean service times; this is shown in the last section of this paper.

2 The matrix-geometric model

We shall denote a state of the system by the tuple (n_a, n_b, x, y), where n_a and n_b respectively represent the number of class A and class B jobs in the system (in the queue or in service). As such, n_a and n_b can take the integer values 0,1,2, ... The indices x and y refer to the class of job that is being served on S_1 and S_2 respectively. Consequently, their values may be A, B or 0; the latter indicates that the respective server is idle. It is necessary to include this information in the state description: since the classes may have unequal mean service times, the class that is being served determines the time at which a particular server becomes idle and as such the class that is served next.

We now order these states lexicographically and construct *levels* $\ell(i)$ of states with an equal number of class A jobs. Within each level, states are grouped according to the number of class B jobs, which we call the *major phase* $\varpi(j)$ of the process. Finally, each such major phase has a number of states corresponding to the values of the indices x and y, the *minor phase* of the process.

The number of minor phases within each major phase may depend on the level. E.g. $\varpi(1)$ within the level $\ell(1)$ has only two minor phases, (A, B) and (B, A), whereas the same major phase within the level $\ell(2)$ has three minor phases, (A, A),(A, B) and (B, A). Hence, the boundary levels $\ell(0)$ and $\ell(1)$ are slightly different from the other *homogeneous* levels. The minor phases for all levels are summarized in Table 1.

We then find that the chain forms a QBD (Quasi-Birth-and-Death) sys-

tem with generator matrix Q given by:

$$Q = \begin{bmatrix} B_{00} & B_{01} & & & & \\ B_{10} & B_{11} & B_{12} & & & \\ & B_{21} & A_1 & A_0 & & \\ & & A_2 & A_1 & A_0 & \\ & & & A_2 & A_1 & \ddots \\ & & & & \ddots & \ddots \end{bmatrix}.$$

Transitions occur between two levels or between two major phases within the same level, with a possible transition in the minor phase at the same time. The behavior of the system at the levels $\ell(i)$ $(i > 2)$, is described in Tables 2 and 3. From these tables, we derive the structure of the matrices A_0, A_1 and A_2. These are square matrices, consisting of rows and columns of blocks corresponding to the major phases. The block $\triangle^*$ denotes a diagonal matrix, the elements of which are such that the row sums of Q equal zero. Empty positions in these matrices indicate that no transitions are possible between the corresponding major phases.

$$A_0 = \begin{bmatrix} L_{A0} & & & \\ & L_{A1} & & \\ & & L_A & \\ & & & \ddots \end{bmatrix},$$

$$A_1 = \begin{bmatrix} \triangle^* & L_{B0} & & & \\ M_{B0} & \triangle^* & L_{B1} & & \\ & M_{B1} & \triangle^* & L_B & \\ & & M_B & \triangle^* & \ddots \\ & & & \ddots & \ddots \end{bmatrix},$$

$$A_2 = \begin{bmatrix} M_{A0} & & & \\ & M_{A1} & & \\ & & M_A & \\ & & & \ddots \end{bmatrix}.$$

The entries of A_0, A_1 and A_2 are blocks; their structure, as well as the structure of the boundary matrices, is displayed in the Appendix.

Notice that these matrices all have an infinite number of entries, since the number of major phases is not bounded by the definition of the queueing

level	$\varpi(0)$	$\varpi(1)$	$\varpi(j), j \geq 2$
$\ell(0)$	$(0,0)$	$(0,B),(B,0)$	(B,B)
$\ell(1)$	$(A,0),(0,A)$	$(A,B),(B,A)$	$(A,B),(B,A),(B,B)$
$\ell(i), i \geq 2$	(A,A)	$(A,A),(A,B),(B,A)$	$(A,A),(A,B),(B,A),(B,B)$

Table 1: Minor phases for each major phase in each level.

	from state	$\rightarrow$	to state	rate	initial condition
level up	(i,j,x,y)	$\rightarrow$	$(i+1,j,x,y)$	λ_a	
level down	(i,j,A,B)	$\rightarrow$	$(i-1,j,A,B)$	μ_a	$j>0$
	(i,j,A,A)	$\rightarrow$	$(i-1,j,A,A)$	$2\mu_a$	$j=0$
		$\rightarrow$	$(i-1,j,A,A)$	μ_a	$j>0$
		$\rightarrow$	$(i-1,j,A,B)$	μ_a	$j>0$
	(i,j,B,A)	$\rightarrow$	$(i-1,j,B,A)$	μ_a	$j=1$
		$\rightarrow$	$(i-1,j,B,B)$	μ_a	$j>1$

Table 2: Transitions between levels.

	from state	$\rightarrow$	to state	rate	initial condition
phase up	(i,j,x,y)	$\rightarrow$	$(i,j+1,x,y)$	λ_b	
phase down	(i,j,A,B)	$\rightarrow$	$(i,j-1,A,A)$	μ_b	$j=1$
		$\rightarrow$	$(i,j-1,A,B)$	μ_b	$j>1$
	(i,j,B,A)	$\rightarrow$	$(i,j-1,A,A)$	μ_b	$j>0$
	(i,j,B,B)	$\rightarrow$	$(i,j-1,A,B)$	μ_b	$j>1$
		$\rightarrow$	$(i,j-1,B,A)$	μ_b	$j=2$
		$\rightarrow$	$(i,j-1,B,B)$	μ_b	$j>2$

Table 3: Transitions within the level.

model. In order to solve the model numerically for the queue length distributions, the matrices should be finite. We shall therefore limit the queue size of class B jobs to M, or, equivalently, we truncate the state space of the process so that n_b, the major phase, cannot exceed M. The matrices A_0, A_1 and A_2 then all consist of $M+1$ rows and columns of blocks.

3 Effect of state space truncation

From Neuts [4], it is clear that the (joint) stationary distribution vector $\boldsymbol{\pi}$ for the number of jobs in the system in this QBD is matrix-geometric, i.e. of the form

$$\boldsymbol{\pi}_i = \boldsymbol{\pi}_2 R^{i-2}, \qquad \forall i \geq 2.$$

Solution of the model proceeds by first determining the value of the rate matrix R (e.g. using the algorithm LR, see Latouche and Ramaswami [2]), then by solving the system of matrix equations

$$\begin{aligned}
\boldsymbol{\pi}_0 B_{00} + \boldsymbol{\pi}_1 B_{10} &= \mathbf{0}, \\
\boldsymbol{\pi}_0 B_{01} + \boldsymbol{\pi}_1 B_{11} + \boldsymbol{\pi}_2 B_{21} &= \mathbf{0}, \\
\boldsymbol{\pi}_1 B_{12} + \boldsymbol{\pi}_2 (A_1 + R A_2) &= \mathbf{0}, \\
\boldsymbol{\pi}_0 \mathbf{1} + \boldsymbol{\pi}_1 \mathbf{1} + \boldsymbol{\pi}_2 (I - R)^{-1} \mathbf{1} &= 1,
\end{aligned}$$

to determine the value of the boundary probability vectors $\boldsymbol{\pi}_0$, $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$. The marginal stationary distributions and their moments are then given as closed form expressions and are easily calculated (see Leemans [3]).

For the numerical examples in this section, we assume that $\mu_a = \mu_b = 1$. The load of the system is varied by choosing appropriate values of λ_a and λ_b. It is clear that with this model, because of the finite value of M, we can only obtain an approximation for the unbounded system. We calculated 'exact' results by increasing the value of M until the average queue lengths (hereafter denoted by N_a and N_b) were stable. For the case $\lambda_a = \lambda_b = 0.95$, the value of M had to be increased to about 200 in order to obtain a stable average queue length for both classes. In the paragraphs below, we only show results for high class A load and varying class B load since here, the average queue lengths are most strongly affected by the truncation.

It is easy to see that truncation results in a lower bound for the average queue lengths of both classes. Whenever the number of class B jobs is equal to M, new class B arrivals are immediately rejected from the system. The effective load of class B is reduced and the average queue length of class B will therefore be smaller than in the unbounded system (primary effect). As

a secondary effect, because there are less class B jobs in the system, more of the capacity is left for class A jobs and they can be served faster, as such reducing class A average queue length. The lower the value of M and the higher the load on class B, the more jobs will be rejected from the system and the larger the impact will be on the average queue lengths. This is clearly seen in Figures 3 and 4. They show the average queue length of class A and B respectively, for a system with class A load $= 0.95$, and class B load varying up to 0.95. The accuracy of the results improves with higher values of M. With $M = 40$, N_a is already close to the 'exact' results. Because of the truncation on class B, N_b is less accurate for the same values of M.

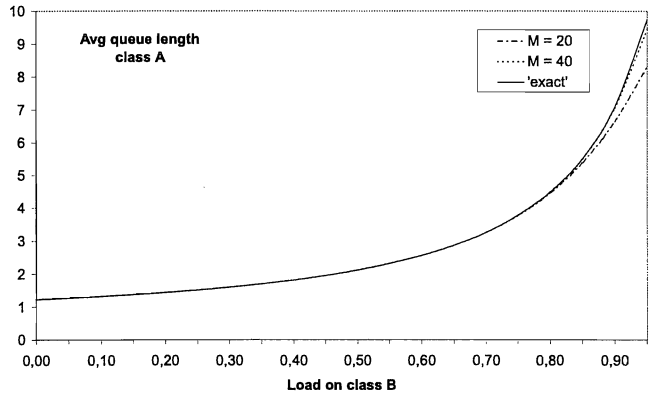

Figure 3: Average queue length for class A, $\rho_a = 0.95$.

4 Proof of the lower bounds

We shall now formally prove that truncation generates a lower bound for the average queue length of both classes. We do so using van Houtum's precedence relation method (See [5] and [6] for details). The method is based on the fact that performance characteristics of a Markov chain can be represented as average costs for an appropriately chosen cost structure. A lower bound is obtained if the chain is modified so that the average costs are lower than in the original chain.

Key step in this method is the determination of precedence pairs between states of the Markov chain; these are pairs of states $(\mathbf{m}, \mathbf{n})$ for which $\mathbf{m}$ is more attractive than $\mathbf{n}$. For the precedence pair $(\mathbf{m}, \mathbf{n})$, it holds that the

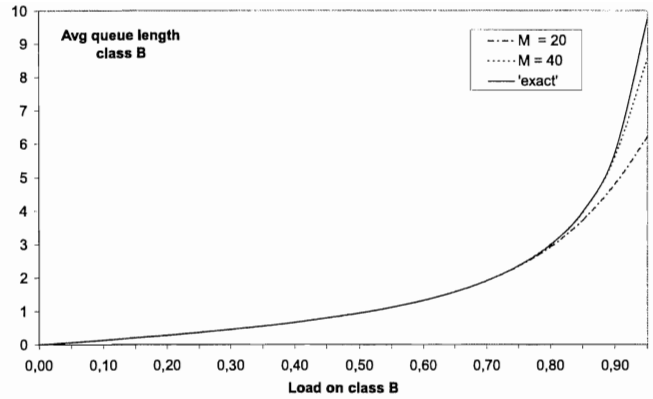

Figure 4: Average queue length for class B, $\rho_a = 0.95$.

total expected cost $v_n(\mathbf{m})$ over n periods of starting in $\mathbf{m}$ is lower than the total expected cost $v_n(\mathbf{n})$ over n periods of starting in $\mathbf{n}$, for all n. If the Markov chain is modified so that states are redirected to more attractive states, then the total expected costs in the modified chain will be lower than in the original model and the average costs will be ordered in the same way.

By truncating the number of class B jobs in our queueing model, we have actually redirected transitions as illustrated in Figure 5. In order to prove the lower bounds for the average queue lengths, we need to show that we redirect transitions to states which are more attractive for both classes. That is, we need to prove the precedence pairs shown in Figure 6.

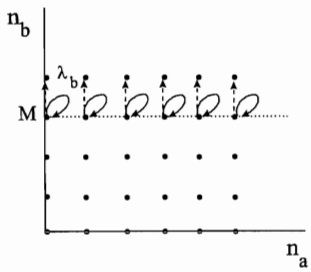

Figure 5: Redirection of transitions by truncation.

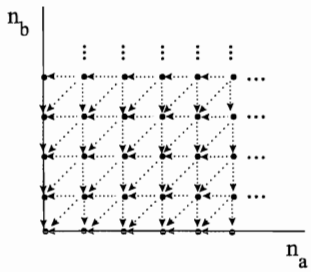

Figure 6: Precedence pairs for both classes. The arrows point in the direction of the more attractive states.

We shall therefore regard the average marginal queue lengths of class A and of class B in our queueing model respectively as the average costs

$$\begin{aligned} g_a &= \sum_{\mathbf{m}} c_a(\mathbf{m})\pi(\mathbf{m}), \\ g_b &= \sum_{\mathbf{m}} c_b(\mathbf{m})\pi(\mathbf{m}), \end{aligned}$$

with one-period costs

$$c_a(\mathbf{m}) = n_a, \tag{1}$$
$$c_b(\mathbf{m}) = n_b, \tag{2}$$

and with states $\mathbf{m} = (n_a, n_b)$. Notice that we now ignore the minor phases of the process, since the one-period cost in a state is determined only by n_a or n_b. However, as we shall see, the minor phase plays an important role in the proof of precedence pairs. We state the proof here for class A only; the proof for class B follows the same line. Let us denote by $\mathbf{e}_1$ the vector (1,0) and by $\mathbf{e}_2$ the vector (0,1). Calculation of expected costs over a number of periods requires that our continuous time Markov chain be translated into a discrete time chain; this is done by uniformization (see Çinlar [1], Theorem 8.4.31). We thus obtain a transition matrix $P = \frac{1}{\theta}Q + I$, where $\theta = \max_i |Q_{ii}|$. Each unit of time in this discrete time process now corresponds to one period. The total expected cost over n periods when starting in state $\mathbf{m}$ is then given by

$$v_n(\mathbf{m}) = c(\mathbf{m}) + \sum_{\mathbf{i}} p(\mathbf{m}, \mathbf{i}) v_{n-1}(\mathbf{i}),$$

where $p(\mathbf{m}, \mathbf{i})$ are the transition probabilities as given in P. We are now ready to state and prove the following theorem:

Theorem 1 *If $\mu_a = \mu_b = \mu$, then the general set of precedence pairs P for both classes is equal to*

$$P = \{(\mathbf{m}, \mathbf{n}) \mid \mathbf{n} = \mathbf{m} + \mathbf{e}_1, \mathbf{n} = \mathbf{m} + \mathbf{e}_2, \mathbf{n} = \mathbf{m} + \mathbf{e}_1 + \mathbf{e}_2\}. \qquad \square$$

Proof Define $v_n(\mathbf{m}; \varphi)$ as the expected cost for class A over n periods when starting in the state $\mathbf{m}$ with the minor phase φ. Let Φ be the set of all possible minor phases. The precedence relations are proved by showing that

$$v_n(\mathbf{m}; \varphi) \leq v_n(\mathbf{n}; \varphi'), \quad \forall (\mathbf{m}, \mathbf{n}) \in P, \forall \varphi, \varphi' \in \Phi \text{ and } \forall n \geq 0. \tag{3}$$

The proof is by induction over n.

For $n = 1$, we have that $v_1(\mathbf{m};\varphi) = c_a(\mathbf{m})$. It follows directly from (1) that

$$c_a(\mathbf{m}) \leq c_a(\mathbf{n}), \qquad \forall(\mathbf{m},\mathbf{n}) \in P. \tag{4}$$

We now assume that $v_n(\mathbf{m};\varphi) \leq v_n(\mathbf{n};\varphi')$ is true $\forall(\mathbf{m},\mathbf{n}) \in P$ and $\forall\varphi,\varphi' \in \Phi$. It has to be shown that

$$v_{n+1}(\mathbf{m};\varphi) \leq v_{n+1}(\mathbf{n};\varphi'), \qquad \forall(\mathbf{m},\mathbf{n}) \in P, \forall\varphi,\varphi' \in \Phi. \tag{5}$$

We will first consider the pair $(\mathbf{m}, \mathbf{m}+\mathbf{e}_2)$. Different cases should be distinguished: $\mathbf{m} = (0,0)$, $\mathbf{m} = (1,0)$, $\mathbf{m} = (n_a,0)$ for $n_a \geq 2$, $\mathbf{m} = (0,1)$, $\mathbf{m} = (0,n_b)$ for $n_b \geq 2$, $\mathbf{m} = (1,1)$, $\mathbf{m} = (1,n_b)$ for $n_b \geq 2$, $\mathbf{m} = (n_a,1)$ for $n_a \geq 2$ and $\mathbf{m} = (n_a,n_b)$ for $n_a, n_b \geq 2$. Since the states which are attainable from $(\mathbf{m};\varphi)$ and $(\mathbf{n};\varphi')$ are explicitly determined by φ and φ', a proof for these cases consists of a proof for all possible values of φ and φ' in turn. Consider the case where $\mathbf{m} = (n_a,n_b)$ with $n_a, n_b \geq 2$. Both φ and φ' can have the four values $(A,A),(A,B),(B,A)$ and (B,B). Let us discuss the case where $\varphi = (A,A)$ and $\varphi' = (A,B)$. The expected costs over $n+1$ periods are given by

$$\begin{aligned} v_{n+1}(n_a,n_b;A,A) &= c_a(n_a,n_b) \\ &+\frac{\lambda_a}{\theta}v_n(n_a+1,n_b;A,A) + \frac{\lambda_b}{\theta}v_n(n_a,n_b+1;A,A) \\ &+\frac{\mu}{\theta}v_n(n_a-1,n_b;\xi) + \frac{\mu}{\theta}v_n(n_a-1,n_b;A,B) \\ &+\left(1-\frac{\lambda_a+\lambda_b+2\mu}{\theta}\right)v_n(n_a,n_b;A,A), \end{aligned} \tag{6}$$

where ξ can be either (A,A) or (B,A), depending on the value of n_a, and

$$\begin{aligned} v_{n+1}(n_a,n_b+1;A,B) &= c_a(n_a,n_b+1) \\ &+\frac{\lambda_a}{\theta}v_n(n_a+1,n_b+1;A,B) + \frac{\lambda_b}{\theta}v_n(n_a,n_b+2;A,B) \\ &+\frac{\mu}{\theta}v_n(n_a-1,n_b+1;A,B) + \frac{\mu}{\theta}v_n(n_a,n_b;A,B) \\ &+\left(1-\frac{\lambda_a+\lambda_b+2\mu}{\theta}\right)v_n(n_a,n_b+1;A,B). \end{aligned} \tag{7}$$

We now compare the corresponding terms in the right hand sides of both equations. The first term of (6) is less than or equal to the first term in (7), by (4). The second term in (6) is less than or equal to the second term in (7), by the induction assumption. The same holds for the other terms. We may thus conclude that $v_{n+1}(\mathbf{m};A,A) \leq v_{n+1}(\mathbf{m}+\mathbf{e}_2;A,B)$

for $\mathbf{m} = (n_a, n_b)$ $(n_a, n_b \geq 2)$. We have to repeat the proof for all other combinations of minor phases. They are completely similar to the one shown above; this tedious enumeration of proofs is therefore omitted. We conclude that $v_{n+1}(\mathbf{m}; \varphi) \leq v_{n+1}(\mathbf{m}+\mathbf{e}_2; \varphi')$ for $\mathbf{m} = (n_a, n_b)$ $(n_a, n_b \geq 2)$, $\forall \varphi, \varphi' \in \Phi$.

Subsequently, we must consider the other cases mentioned above. They will slightly differ in the reachable states, but lead us to similar conclusions and, therefore, they are not shown here. We conclude that $v_{n+1}(\mathbf{m}; \varphi) \leq v_{n+1}(\mathbf{m} + \mathbf{e}_2; \varphi')$ for all $\mathbf{m}$ and for all $\varphi, \varphi' \in \Phi$.

Next, we consider the pair $(\mathbf{m}, \mathbf{m}+\mathbf{e}_1)$. We distinguish the same cases as for $(\mathbf{m}, \mathbf{m} + \mathbf{e}_2)$. Again, a proof for all values of φ and φ' is required. We show the proof for $\mathbf{m} = (n_a, 0)$ with $n_a \geq 2$, for which $\Phi = (A, A)$. The expected costs over $n+1$ periods are given by

$$\begin{aligned} v_{n+1}(n_a, 0; A, A) &= c_a(n_a, 0) \\ &+\frac{\lambda_a}{\theta} v_n(n_a + 1, 0; A, A) + \frac{\lambda_b}{\theta} v_n(n_a, 1; A, A) \\ &+\frac{\mu}{\theta} v_n(n_a - 1, 0; \xi) + \frac{\mu}{\theta} v_n(n_a - 1, 0; \zeta) \\ &+\left(1 - \frac{\lambda_a + \lambda_b + 2\mu}{\theta}\right) v_n(n_a, 0; A, A), \end{aligned}$$

where ξ and ζ can be either both (A, A), or $(A, 0)$ and $(0, A)$ respectively, depending on the value of n_a, and

$$\begin{aligned} v_{n+1}(n_a + 1, 0; A, A) &= c_a(n_a + 1, 0) \\ &+\frac{\lambda_a}{\theta} v_n(n_a + 2, 0; A, A) + \frac{\lambda_b}{\theta} v_n(n_a + 1, 1; A, A) \\ &+\frac{\mu}{\theta} v_n(n_a, 0; A, A) + \frac{\mu}{\theta} v_n(n_a, 0; A, A) \\ &+\left(1 - \frac{\lambda_a + \lambda_b + 2\mu}{\theta}\right) v_n(n_a + 1, 0; A, A). \end{aligned}$$

It is directly seen that $v_{n+1}(n_a, 0; A, A) \leq v_{n+1}(n_a + 1, 0; A, A)$ $(n_a \geq 2)$ by pairwise comparison of the terms in the right hand sides. Again, the proofs for all other minor phases and for all other cases are similar and we conclude that $v_{n+1}(\mathbf{m}; \varphi) \leq v_{n+1}(\mathbf{m} + \mathbf{e}_1; \varphi')$ for all $\mathbf{m}$ and for all $\varphi, \varphi' \in \Phi$.

The proof of (5) for $(\mathbf{m}, \mathbf{m} + \mathbf{e}_1 + \mathbf{e}_2)$ is based on the transitivity property of $\leq$: if (5) holds for $(\mathbf{m}, \mathbf{m} + \mathbf{e}_1)$, $\forall \mathbf{m}, \forall \varphi, \varphi' \in \Phi$, and for $(\mathbf{m}, \mathbf{m} + \mathbf{e}_2)$, $\forall \mathbf{m}, \forall \varphi, \varphi' \in \Phi$, then it also holds for $(\mathbf{m}, \mathbf{m} + \mathbf{e}_1 + \mathbf{e}_2)$, $\forall \mathbf{m}, \forall \varphi, \varphi' \in \Phi$.

Since (5) holds for all $(\mathbf{m}, \mathbf{n}) \in P$ and $\forall \varphi, \varphi' \in \Phi$, by induction, (3) holds $\forall n$, which completes our proof. □

5 Generation of upper bounds

In the previous section, we have shown that truncation leads to lower bounds for both classes (if they have equal mean service times). These lower bounds are flexible in the sense that they can be made arbitrarily accurate at the cost of increasing computational effort. Indeed, increasing the value of the truncation parameter M improves the accuracy of the average queue length, but also increases the matrix dimensions and therefore CPU time and memory requirements. It would now be interesting to have flexible upper bounds as well, so that the 'exact' average is enclosed in an interval which can be made arbitrarily narrow.

It is immediately clear that an upper bound for the class on the major phase cannot be directly obtained, because truncation on the major phase is necessary for numerical computations and this only leads to lower bounds. Let us therefore define and prove additional sets of precedence pairs for both classes in the following theorems, which hold if $\mu_a = \mu_b = \mu$. These precedence pairs are illustrated in Figures 7 and 8.

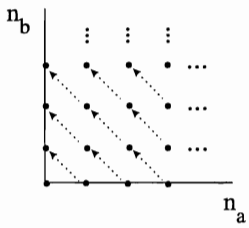

Figure 7: Additional precedence pairs for class A.

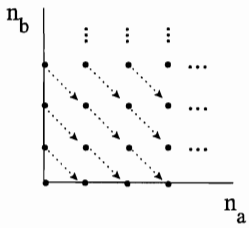

Figure 8: Additional precedence pairs for class B.

Theorem 2 *If $\mu_a = \mu_b = \mu$, then the pair $(\mathbf{m}, \mathbf{m} + \mathbf{e}_1 - \mathbf{e}_2)$ belongs to the set of precedence pairs for class A.* □

Proof Again, the proof is established by induction over n.

For $n = 1$, it follows directly from (1) that

$$c_a(\mathbf{m}) \leq c_a(\mathbf{m} + \mathbf{e}_1 - \mathbf{e}_2), \qquad \forall \mathbf{m}. \tag{8}$$

We now assume that $\upsilon_n(\mathbf{m};\varphi) \leq \upsilon_n(\mathbf{m}+\mathbf{e}_1-\mathbf{e}_2;\varphi')$ is true $\forall\varphi,\varphi' \in \Phi$. We must then show that

$$\upsilon_{n+1}(\mathbf{m};\varphi) \leq \upsilon_{n+1}(\mathbf{m}+\mathbf{e}_1-\mathbf{e}_2;\varphi'), \quad \forall\mathbf{m},\forall\varphi,\varphi' \in \Phi. \tag{9}$$

Again, different cases should be distinguished: $\mathbf{m} = (0,1)$, $\mathbf{m} = (n_a,1)$ for $n_a \geq 1$, $\mathbf{m} = (0,n_b)$ for $n_b \geq 2$ and $\mathbf{m} = (n_a,n_b)$ for $n_a, n_b \geq 2$. Consider the case with $\mathbf{m} = (0,1)$, with $\varphi = (0,B)$ and $\varphi' = (A,0)$. The expected costs over $n+1$ periods are given by

$$\begin{aligned}\upsilon_{n+1}(0,1;0,B) &= c_a(0,1)\\ &+\frac{\lambda_a}{\theta}\upsilon_n(1,1;A,B) + \frac{\lambda_b}{\theta}\upsilon_n(0,2;B,B)\\ &+\frac{\mu}{\theta}\upsilon_n(0,0;0,0) + \left(1-\frac{\lambda_a+\lambda_b+\mu}{\theta}\right)\upsilon_n(0,1;0,B),\end{aligned}$$

$$\begin{aligned}\upsilon_{n+1}(1,0;A,0) &= c_a(1,0)\\ &+\frac{\lambda_a}{\theta}\upsilon_n(2,0;A,A) + \frac{\lambda_b}{\theta}\upsilon_n(1,1;A,B)\\ &+\frac{\mu}{\theta}\upsilon_n(0,0;0,0) + \left(1-\frac{\lambda_a+\lambda_b+\mu}{\theta}\right)\upsilon_n(1,0;A,0).\end{aligned}$$

Comparing the corresponding terms in the right hand sides of both equations, it is clear that the first terms are ordered by (8). The second, third and fifth terms are ordered by the induction assumption, while the fourth terms are exactly equal because of the equal mean service rates. The proof for this case with other minor phases is similar. Therefore, we conclude that $\upsilon_{n+1}(\mathbf{m};\varphi) \leq \upsilon_{n+1}(\mathbf{m}+\mathbf{e}_1-\mathbf{e}_2;\varphi')$ for $\mathbf{m} = (0,1)$ $\forall\varphi,\varphi' \in \Phi$.

Similarly, we may establish the proof for the other cases. E.g. for $\mathbf{m} = (n_a,n_b)$ $(n_a,n_b \geq 2)$, with $\varphi = (A,A)$ and $\varphi' = (A,B)$, the expected costs are

$$\begin{aligned}\upsilon_{n+1}(n_a,n_b;A,A) &= c_a(n_a,n_b)\\ &+\frac{\lambda_a}{\theta}\upsilon_n(n_a+1,n_b;A,A) + \frac{\lambda_b}{\theta}\upsilon_n(n_a,n_b+1;A,A)\\ &+\frac{\mu}{\theta}\upsilon_n(n_a-1,n_b;\xi) + \frac{\mu}{\theta}\upsilon_n(n_a-1,n_b;A,B)\\ &+\left(1-\frac{\lambda_a+\lambda_b+2\mu}{\theta}\right)\upsilon_n(n_a,n_b;A,A),\end{aligned}$$

where ξ can be (A, A) or (B, A), depending on the value of n_a, and

$$\begin{aligned}
\upsilon_{n+1}(n_a+1, n_b-1; A, B) &= c_a(n_a+1, n_b-1) \\
&+\frac{\lambda_a}{\theta}\upsilon_n(n_a+2, n_b-1; A, B) + \frac{\lambda_b}{\theta}\upsilon_n(n_a+1, n_b; A, B) \\
&+\frac{\mu}{\theta}\upsilon_n(n_a, n_b-1; A, B) + \frac{\mu}{\theta}\upsilon_n(n_a+1, n_b-2; \zeta) \\
&+\left(1 - \frac{\lambda_a+\lambda_b+2\mu}{\theta}\right)\upsilon_n(n_a+1, n_b-1; A, B),
\end{aligned}$$

where ζ can be (A, B) or (A, A), depending on the value of n_b. By pairwise comparison of the terms in the right hand sides, and assuming that the induction assumption holds, it is directly seen that $\upsilon_{n+1}(n_a, n_b; A, A) \leq \upsilon_{n+1}(n_a+1, n_b-1; A, B)$ $(n_a, n_b \geq 2)$. The proof for the other minor phases is similar. These proofs, as well as those for the other cases, are therefore omitted and we conclude that $(\mathbf{m}, \mathbf{m}+\mathbf{e}_1-\mathbf{e}_2)$ belongs to the set of precedence pairs for class A. □

Finally, we also state the following theorem.

Theorem 3 *If $\mu_a = \mu_b = \mu$, then the pair $(\mathbf{m}, \mathbf{m}-\mathbf{e}_1+\mathbf{e}_2)$ belongs to the set of precedence pairs for class B.* □

Proof The proof is immediately found by using (2) as the cost function and proceeding similarly as for Theorem 2. □

We now obtain an upper bound for class A (the class on the level of the process) by the following modification:

> *If M class B jobs are present in the system, a newly arriving class B job is added to the queue of class A and is from then on treated as a class A job.*

This modification is graphically represented in Figure 9. A transition from (n_a, M) to $(n_a, M+1)$ is redirected to (n_a+1, M), which, by Theorem 2 is proved to be a less attractive state for class A if the mean service rates are equal. Simultaneously, Theorem 3 states that this is a more attractive state for class B under these circumstances. Hence, we obtain at the same time an upper bound for class A and a lower bound for class B.

The use of matrix-geometric methods facilitates the analysis of this modified model. The proposed modification only involves minor changes to the matrices representing transitions to a higher level (the matrices A_0, B_{01} and B_{12}). For $n_b = M$, a transition to the next higher level is now not only due

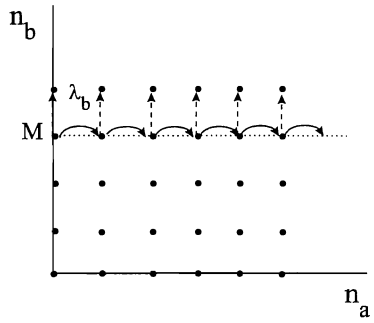

Figure 9: Generation of upper bounds for class A by modification of CBM.

to the arrival of a class A job, but also to the arrival of a class B job, which is transformed into a class A job. The transition rate for these states changes from λ_a to $\lambda_a + \lambda_b$.

We may now also obtain an upper bound for class B (the class on the major phase) by analyzing the reversed model, in which level and phase of the process are switched.

Figure 10 illustrates the bounds for class A, for the same class loads as in Figure 3 with $M = 20$. If we compare the upper and lower bounds resulting from the modification, we observe from the numerical examples that the upper bound is tighter than the lower bound for values of ρ_b in the range 0.55 to 0.90. The upper bound results from an artificial increase of class A load at any occasion where the arrival of a class B job causes the treshold M to be exceeded. This will not occur very frequently at moderate class B loads. On the other hand, the lower bound at these values is obtained by a reduction of class A load, as a direct consequence of the truncation of this class in the reversed model. This effect is quite strong for high class A loads and is reinforced by increasing class B loads. The lower bound resulting from the modification is therefore also less tight than the lower bound resulting from the original truncation, which was basically generated by the indirect effect of a lowered effective class B load. Of course, these bounds improve as M increases.

The bounds for class B are shown in Figure 11. The lower bounds obtained by the original truncation are almost coinciding with those from the modification; it is clear how the redirection of a transition from $(n_a, M+1)$ to $(n_a + 1, M)$, instead of to (n_a, M), adversely affects the lower bound for class B, so that this new bound is slightly tighter. The upper bound for class B is now generated by the reversed model – with class A being the phase –

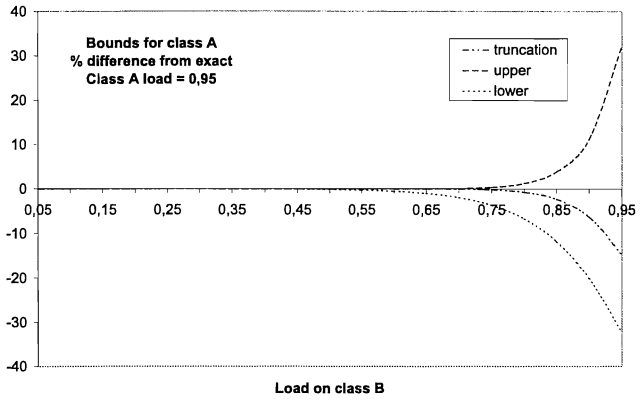

Figure 10: Comparison of bounds for N_a ($\rho_a = 0.95, M = 20$ and equal mean service times).

and is strongly affected by M because of the high class A load.

6 The case of unequal mean service times

It appears to be intuitively clear that Theorem 1 should hold for the case where the classes have unequal mean service times, since both classes benefit if the system has fewer jobs of either of the classes, no matter what the service rates are. However, the proof technique does not allow us to prove the precedence pairs in this case. The problem is caused by the presence of minor phases; although they do not determine the one-period cost, they have a considerable influence on the path of states that is followed through the chain. In order to compare the expected costs of two states, it is necessary to compare all states on all paths which may be followed when starting from these two states. Most systems for which bounds have been proved using this technique (see van Houtum [6]) are single class systems with a two-dimensional state space, which is such that the paths which are followed when starting in two adjacent states mostly coincide. In our model, the minor phase adds a third dimension to the state space and determines the outgoing rates to adjacent states. As it may change due to completions of jobs of both classes, the paths which are followed when starting from two adjacent states may be different, thus largely complicating the proof of precedence pairs.

To illustrate where the difficulty arises, we recall (6) and (7) in the proof of Theorem 1. The comparison of the fifth term in the right hand sides of

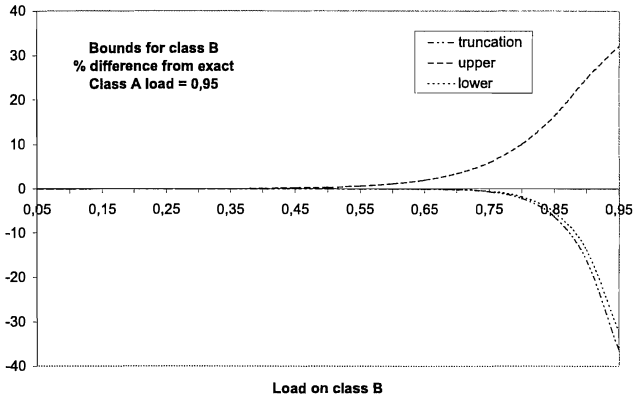

Figure 11: Comparison of bounds for N_b ($\rho_a = 0.95, M = 20$ and equal mean service times).

these expressions explicitly relies on the assumption that the mean service rates are equal. In (6), the minor phase in the left hand side is (A, A), so that completions are due to class A only. In (7), completions are due to both classes. With unequal mean service rates, the fifth terms in the right hand sides of both expressions respectively become

$$\frac{\mu_a}{\theta} \upsilon_n(n_a - 1, n_b; A, B)$$

and

$$\frac{\mu_b}{\theta} \upsilon_n(n_a, n_b; A, B).$$

It follows from the induction assumption that

$$\upsilon_n(n_a - 1, n_b; A, B) \leq \upsilon_n(n_a, n_b; A, B).$$

However, due to unequal mean service rates, it may happen that

$$\frac{\mu_a}{\theta} \upsilon_n(n_a - 1, n_b; A, B) > \frac{\mu_b}{\theta} \upsilon_n(n_a, n_b; A, B).$$

Therefore, we cannot, with this approach, prove that $\upsilon_{n+1}(n_a, n_b; A, A) \leq \upsilon_{n+1}(n_a, n_b + 1; A, B)$ if the service rates differ.

The same problems arise in proving Theorems 2 and 3 for unequal mean service times. It is however possible to obtain similar bounds as in the

previous section for the case where the classes have unequal mean service rates. As such, we found that the modification also results in an upper bound for class A and a lower bound for class B if $\mu_b = 2\mu_a$, so that both Theorem 2 and 3 still hold for this case. However, for $\mu_b = 10\mu_a$, we find upper bounds for *both* classes, which indicates that Theorem 3 is no longer valid. For $\mu_a = 10\mu_b$, we find lower bounds for *both* classes, meaning that Theorem 2 is now invalid. It thus seems that the theorems are valid as long as the difference between the mean service rates is not too large. However, we cannot prove this with the precedence relation method, nor does the technique allow us to quantify the region of parameter values in which the theorems would be valid.

Appendix

The subblocks of A_0, A_1 and A_2 are shown below. Within each block, the minor phases are ordered lexicographically, as in Table 1. To make the non-zero elements clearly stand out, we indicate zeroes in these matrices with a dot. I_n denotes an identity matrix of size n.

$$L_{A0} = \lambda_a, \qquad L_{A1} = \lambda_a I_3, \qquad L_A = \lambda_a I_4,$$

$$L_{B0} = \begin{bmatrix} \lambda_b & \cdot & \cdot \end{bmatrix}, \quad L_{B1} = \begin{bmatrix} \lambda_b & \cdot & \cdot & \cdot \\ \cdot & \lambda_b & \cdot & \cdot \\ \cdot & \cdot & \lambda_b & \cdot \end{bmatrix}, \quad L_B = \lambda_b I_4,$$

$$M_{A0} = 2\mu_a, \qquad M_{A1} = \begin{bmatrix} \mu_a & \mu_a & \cdot \\ \cdot & \mu_a & \cdot \\ \cdot & \cdot & \mu_a \end{bmatrix}, \qquad M_A = \begin{bmatrix} \mu_a & \mu_a & \cdot & \cdot \\ \cdot & \mu_a & \cdot & \cdot \\ \cdot & \cdot & \cdot & \mu_a \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix},$$

$$M_{B0} = \begin{bmatrix} \cdot \\ \mu_b \\ \mu_b \end{bmatrix}, \qquad M_{B1} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \mu_b & \cdot \\ \mu_b & \cdot & \cdot \\ \cdot & \mu_b & \mu_b \end{bmatrix}, \qquad M_B = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \mu_b & \cdot & \cdot \\ \mu_b & \cdot & \cdot & \cdot \\ \cdot & \mu_b & \cdot & \mu_b \end{bmatrix}.$$

The boundary matrices have the same global structure as the A-matrices; their subblocks have, however, other dimensions because of the different number of minor phases in each major phase (see again Table 1). The structure is shown below. The order of the states in each block is lexicographically and corresponds to the one reported in Table 1. The $\triangle^*$ on the diagonal of B_{00} and B_{11} again indicates an appropriately sized diagonal matrix, the elements of which are such that the row sums of Q equal zero.

$$B_{00} = \begin{bmatrix} \triangle^* & L_{00}^{(0)} & & & \\ M_{00}^{(0)} & \triangle^* & L_{00}^{(1)} & & \\ & M_{00}^{(1)} & \triangle^* & L_{00} & \\ & & M_{00} & \triangle^* & \ddots \\ & & & \ddots & \ddots \end{bmatrix}, \quad B_{01} = \begin{bmatrix} L_{01}^{(0)} & & & \\ & L_{01}^{(1)} & & \\ & & L_{01} & \\ & & & \ddots \end{bmatrix},$$

$$B_{10} = \begin{bmatrix} M_{10}^{(0)} & & & \\ & M_{10}^{(1)} & & \\ & & M_{10} & \\ & & & \ddots \end{bmatrix}, \quad B_{11} = \begin{bmatrix} \triangle^* & L_{11}^{(0)} & & & \\ M_{11}^{(0)} & \triangle^* & L_{11}^{(1)} & & \\ & M_{11}^{(1)} & \triangle^* & L_{11} & \\ & & M_{11} & \triangle^* & \ddots \\ & & & \ddots & \ddots \end{bmatrix},$$

$$B_{12} = \begin{bmatrix} L_{12}^{(0)} & & & \\ & L_{12}^{(1)} & & \\ & & L_{12} & \\ & & & \ddots \end{bmatrix}, \quad B_{21} = \begin{bmatrix} M_{21}^{(0)} & & & \\ & M_{21}^{(1)} & & \\ & & M_{21} & \\ & & & \ddots \end{bmatrix},$$

with

$$L_{00}^{(0)} = \begin{bmatrix} \lambda_b & \cdot \end{bmatrix}, \quad L_{00}^{(1)} = \begin{bmatrix} \lambda_b \\ \lambda_b \end{bmatrix}, \qquad L_{00} = \lambda_b,$$

$$L_{01}^{(0)} = \begin{bmatrix} \lambda_a & \cdot \end{bmatrix}, \quad L_{01}^{(1)} = \lambda_a I_2, \qquad L_{01} = \begin{bmatrix} \cdot & \cdot & \lambda_a \end{bmatrix},$$

$$L_{11}^{(0)} = \lambda_b I_2, \qquad L_{11}^{(1)} = \begin{bmatrix} \lambda_b & \cdot & \cdot \\ \cdot & \lambda_b & \cdot \end{bmatrix}, \quad L_{11} = \lambda_b I_3,$$

$$L_{12}^{(0)} = \begin{bmatrix} \lambda_a \\ \lambda_a \end{bmatrix}, \qquad L_{12}^{(1)} = \begin{bmatrix} \cdot & \lambda_a & \cdot \\ \cdot & \cdot & \lambda_a \end{bmatrix}, \quad L_{12} = \begin{bmatrix} \cdot & \lambda_a & \cdot & \cdot \\ \cdot & \cdot & \lambda_a & \cdot \\ \cdot & \cdot & \cdot & \lambda_a \end{bmatrix}$$

and

$$M_{00}^{(0)} = \begin{bmatrix} \mu_b \\ \mu_b \end{bmatrix}, \qquad M_{00}^{(1)} = \begin{bmatrix} \mu_b & \mu_b \end{bmatrix}, \qquad M_{00} = 2\mu_b,$$

$$M_{10}^{(0)} = \begin{bmatrix} \mu_a \\ \mu_a \end{bmatrix}, \qquad M_{10}^{(1)} = \mu_a I_2, \qquad M_{10} = \begin{bmatrix} \mu_a \\ \mu_a \\ \cdot \end{bmatrix},$$

$$M_{11}^{(0)} = \mu_b I_2, \qquad M_{11}^{(1)} = \begin{bmatrix} \mu_b & \cdot \\ \cdot & \mu_b \\ \mu_b & \mu_b \end{bmatrix}, \qquad M_{11} = \begin{bmatrix} \mu_b & \cdot & \cdot \\ \cdot & \mu_b & \cdot \\ \mu_b & \cdot & \mu_b \end{bmatrix},$$

$$M_{21}^{(0)} = \begin{bmatrix} \mu_a & \mu_a \end{bmatrix}, \quad M_{21}^{(1)} = \begin{bmatrix} \mu_a & \mu_a \\ \mu_a & \cdot \\ \cdot & \mu_a \end{bmatrix}, \quad M_{21} = \begin{bmatrix} \mu_a & \mu_a & \cdot \\ \mu_a & \cdot & \cdot \\ \cdot & \cdot & \mu_a \\ \cdot & \cdot & \cdot \end{bmatrix}.$$

References

[1] E. Çinlar. *Introduction to Stochastic Processes.* Prentice-Hall, Inc., Englewood Cliffs, 1975.

[2] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for Quasi-Birth-Death processes. *J. Appl. Prob.*, 30:650–674, 1993.

[3] H. Leemans. *The Two-Class Two-Server Queueing Model with Nonpreemptive Heterogeneous Priority Structures.* PhD thesis, K.U.Leuven, Department of Applied Economic Sciences, 1998.

[4] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach.* The John Hopkins University Press, Baltimore, Md., 1981.

[5] G. J. van Houtum. *New Approaches for Multi-Dimensional Queueing Systems.* PhD thesis, Technische Universiteit Eindhoven, 1995.

[6] G. J. van Houtum, W. H. M. Zijm, I. J. B. F. Adan, and J. Wessels. Bounds for performance characteristics; a systematic approach via cost structures. *Commun. Statist. — Stochastic Models*, 14:205–224, 1998.